UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/027,359	12/19/2001	Fay Chong JR.	5681-05200	9639

7590 06/20/2006

Robert C. Kowert
Conley, Rose, & Tayon, P.C.
P.O. Box 398
Austin, TX 78767

EXAMINER
TSAI, SHENG JEN

ART UNIT	PAPER NUMBER
2186	

DATE MAILED: 06/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/027,359	**Applicant(s)** CHONG, FAY	
	Examiner Sheng-Jen Tsai	**Art Unit** 2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 May 2006*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) *1,10,16 and 27* is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *2-9, 11-15, 17-26 and 28-34* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This Office Action is taken in response to Applicants' Amendments and Remarks filed on May 24, 2006 regarding application 10/027,359 filed on December 19, 2001.

2. Claims 1, 10, 16, and 27 have been cancelled previously.

Claims 28 and 32-34 have been amended.

Claims 2-9, 11-15, 17-26 and 28-34 are pending for consideration.

3. ***Examiner's Response to Amendments and Remarks***

Applicants' amendments and remarks have been fully and carefully considered.

In response, another iteration of claim analysis based on the previously cited prior art Crater et al. (US 5,146,588) has been embarked. Refer to the corresponding sections of the claim analysis for details.

Independent claims 28 and 32-34 each has been amended with additional limitations of "a memory configured to provided an addressable block operand storage space and to store block operands within the block operand storage space;" "wherein the plurality of block storage locations are configured to cache a portion of the block operand storage space of the memory and ... , such that during the first accumulation operation, the plurality of block storage locations are concurrently configured both to cache certain ones of the block operands and to accumulate the immediate result of the first accumulation operation."

Further, Applicants contend that Crater et al. (US 5,146,588) do not teach the amended limitations because Crater et al. teach the implementation of caching and accumulation functions within separate, non-overlapping structure within cache memory (figure 3, 113) while the storage of intermediate accumulation results is performed within the redundancy accumulator (figure 4, 301), and the redundancy accumulator is not a cache of any memory.

The Examiner disagrees with this assessment for the following reason.

First, in Crater's invention, two levels of caching are provided for the memory system.

The first level of caching is provide by the memory elements (figure 3, 340~355; column 7, lines 5-20) of the cache memory unit shown in figure 1, 113, figure 2, 113 and figure 3. The memory elements (figure 3, 340~355) serves as a cache (figure 1, 113) between a mass memory entity comprising a plurality of storage devices (disk drive units, figure 1, 122-l~125-r), and a plurality of host processors (figure 1, 11~12), and all data transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache accumulator memory (column 6, 7-10). In other words, cache memory (113) serves as a cache of the storage devices (figure 1, 122-l~125-r).

The second level of caching is provide by the redundancy accumulator memory (figure 4, 301), which resides inside the cache controller (figure 3, 331 and 332) and serves as a cache between a plurality of cache memory elements (figure 3, 340~355) and a plurality of interface units (figure 3, 361~364). This is the case because one of the main object of Crater et al.'s invention is to provide reliability by performing redundancy calculation of the data to form a redundancy group in the disk drive subsets. To achieve this goal, not only that all data-transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory 113 (column 6, 7-10), the data must also go through the redundancy accumulator memory 301 (figures 3 and 4) so that the parity can be calculated [the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57)]. In other words, the redundancy accumulator memory (figure 4, 301) serves as a cache of cache memory elements (figure 3, 340~355).

Second, the plurality of block storage locations of the redundancy accumulator memory (figure 4, 301) are concurrently configured both to cache certain ones of the block operands and to accumulate the immediate result of the first accumulation operation [When valid prior data or prior redundancy products are stored in the addressed memory location of redundancy accumulator memory 301, these are used by redundancy generator 305 to combine with the received data byte from latch 303 in the present physical track to generate a new redundancy product which is written into redundancy accumulator memory 301 as part of the well known redundancy calculation. As each data byte is read from the presently received physical track and transmitted through latch 303 to redundancy generator 305 and multiplexer 304, this data is either directly written into the addressed memory location in redundancy accumulator memory 301 or used to perform a redundancy calculation with the prior product stored therein and this new product is then written into the addressed memory location (column 9, lines 55-63); This requires the use of redundancy accumulator memory 301 which is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 53-57)].

Therefore, Crater et al. clearly demonstrate and teach the amended limitations recited in these claims, and the Examiner's position regarding the status of these claims, and all those claims depending from it, remains the same as stated in the previous Office action.

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

5. Claims 2-9, 11-15, 17-26 and 28-34 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being anticipated by claims 1-36 of copending Application No. **10/027,353**, as shown in the following table. Although not all of the conflicting claims are exactly identical, they are extremely similar and are not patentably distinct from each other as explained in the "explanation" column of the table below:

10/027,359 (amended as of 5/31/2005)	**10/027,353 (amended as of 3/11/2005)**	**EXPLANATION**
32	1, 34	Both describe similar apparatus with similar features/functions such as storage array, cache accumulator memory, block operations, functional unit, and the intermediate result being both a result and an operand of the accumulation operation.
33	1, 20	Both describe similar method of using similar features/functions such as storage array, cache accumulator memory, block operations, functional unit, and the intermediate result being both a result and an operand of the accumulation operation.
34	1, 33	Both describe similar means of providing similar features/functions such as storage array, cache accumulator memory, block operations, functional unit, and the intermediate result being both a result and an operand of the accumulation operation.
2	2	Both recite the cache memory being a dual-ported memory.
3	3	Both recite that cache memory comprises at least two independently interfaced memory banks.
4	4	Both recite that the cache is configured to indicate whether a particular block stored in the cache is modified with respect to a copy in main memory.
5	5	Both recite that the cache is to load a copy of operand from memory if it is not present in the cache.
6	6	Both recite that if all the block storage locations in the cache are currently storing valid data, the cache is to select one of the block storage location for overwriting.
7	7	Both recite the use of the least recently used algorithm to overwrite.
8	8	Both recite writing data back to memory before loading the copy to the selected storage location in cache.
9	13	Both recite the functional unit is to perform parity calculation on the block operands.
11	15	Both recite the first block operand is a first one of the data blocks in the stripe of data.
12	16	Both recite the functional unit is to perform the operation on two block-operands.
13	17	Both recite the same sources of the first and the second operands.
14	18	Both recite the same sources of the first and the second operands.
15	19	Both recite the cache is to store a word of the block result during an access cycle in which the cache is also to provide a word of block operand to the functional unit.
17	19	Both recite the cache is to store a word of the block result during an access cycle in which the cache is also to provide a word of block operand to the functional unit.

18	21	Both recite the cache memory being a dual-ported memory.
19	22	Both recite that cache memory comprises at least two independently interfaced memory banks.
20	23	Both recite that if all the block storage locations in the cache are currently storing valid data, the cache is to select one of the block storage location for overwriting.
21	24	Both recite the use of the least recently used algorithm to overwrite.
22	25	Both recite writing data back to memory if the data is modified with respect to a copy stored in the memory.
23	28	Both recite the functional unit is to perform parity calculation on the block operands and generate block results.
24	29	Both recite the command is issued by a storage system controller.
25	30	Both recite the functional unit is to perform the operation on two block-operands.
26	31	Both recite the second operand is to be provided from a data bus.
28	34	Both describe similar data processing system performing similar operations.
29	35	Both recite the same sources of the first and the second operands.
30	36	Both recite the same sources of the first and the second operands.
31	19	Both recite the cache is to store a word of the block result during an access cycle in which the cache is also to provide a word of block operand to the functional unit.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 2, 4-6, 9, 11-18, 20, 23-26 and 28-34 are rejected under 35 U.S.C. 102(b) as being anticipated by Crater et al., (US 5,146,588).

As to claim 32, Crater et al. disclose **a system** [Redundancy Accumulator for Disk Drive Array Memory (title)], **comprising**:

a storage array including a plurality of mass storage devices [data storage subsystem (figure 1, 100) comprising a plurality of disk drives (figure1, 103-l)]; **and an array controller** [control unit (figure 1, 101), and control and drive circuits (figure 1, 121)] **configured to perform block operations on data stored to the storage array**

[the corresponding block operations are the generation of redundancy data using a redundancy accumulator memory (column 2, lines 3-44; column 7, lines 23-68; column 8, lines 1-15; column 8, lines 55-68; column 9, lines 1-68; column 10, lines 1-9); the redundancy accumulator memory accumulate data over a number of data elements, such as records, sectors or tracks (column 7, lines 42-45), hence performing block operations], **wherein the array controller includes a memory** [the corresponding memory is the memory elements (figure 3, 340~355; column 7, lines 5-20) of the cache memory unit shown in figure 1, 113, figure 2, 113 and figure 3] **configured to provided an addressable block operand storage space and to store block operands within the block operand storage space block operands received from one or more of said plurality of mass storage devices** [the memory elements (figure 3, 340~355; column 7, lines 5-20) serves as a cache (figure 1, 113) between a mass memory entity comprising a plurality of storage devices (disk drive units, figure 1, 122-l~125-r), and a plurality of host processors (figure 1, 11~12), and all data transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory (column 6, 7-10) to provide the operands of the redundancy accumulation calculations]**; and a cache accumulator memory** [the corresponding cache accumulator memory is the redundancy accumulator memory (figure 4, 301), which resides inside the cache controller (figure 3, 331 and 332) and serves as a cache between a plurality of cache memory elements (figure 3, 340~355) and a plurality of interface units (figure 3, 361~364). This is the case because one of the main object of Crater et al.'s invention is to provide reliability by performing redundancy calculation of the data to form a redundancy group in the disk drive subsets. To achieve this goal, not only that all data-transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory 113 (column 6, 7-10), the data must also go through the redundancy accumulator memory 301 (figures 3 and 4) so that the parity can be calculated [the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57)]. In other words, the redundancy accumulator memory (figure 4, 301) serves as a cache of cache memory elements (figure 3, 340~355)] **comprising a plurality of block storage locations** [the redundancy accumulator memory is typically an n by k memory (column 8, lines 23-24); corresponding to the N disk drive containing the payload data; figure 4 shows that the redundancy accumulator memory may provide an operand where the corresponding data path is OUT port of the redundancy accumulator (301) → LATCH (306) → multiplexer (306) → IN port of the redundancy accumulator (301)] **and a functional unit** [the redundancy calculator/generator, figure 4; column 7, lines 22-68; column 8, lines 1-15] **configured to perform a block operation** [the redundancy data generations such as simple parity, orthogonal parity, or Reed-Solomon code (column 7, lines 38-41)] **on one or more block operands to generate a block result** [the redundancy accumulator memory accumulate data over a number of data elements, such as records, sectors or tracks (column 7, lines 42-45), hence performing block operations; redundancy accumulator memory is typically an n by k memory (column 8, lines 23-24); the cache memory unit (figure 3, 113), including the memory elements (figure 3, 340~355) and the cache memory controllers (figure 3, 331~332) where the redundancy accumulator memory resides (figure 4, 301), serves as a cache memory between a mass memory entity comprising a plurality of storage devices (disk drive units, figure 1, 122-l~125-r), and a plurality of host processors (figure 1, 11~12)]; **wherein the plurality of block storage locations are configured to cache a portion of the block operand storage space of the memory** [the redundancy accumulator memory (figure 4, 301), which resides inside the cache controller (figure 3, 331 and 332) and serves as a cache between a plurality of cache memory elements (figure 3, 340~355) and a plurality of interface units (figure 3, 361~364). This is the case because one of the main object of Crater et al.'s invention is to provide reliability by performing redundancy calculation of the data to form a redundancy group in the disk drive subsets. To achieve this goal, not only that all data-transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory 113 (column 6, 7-10), the data must also go through the redundancy accumulator memory 301 (figures 3 and 4) so that the parity can be calculated [the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57)]. In other words, the redundancy accumulator memory (figure 4, 301) serves as a cache of cache memory elements (figure 3, 340~355)];

wherein in response to an instruction [instructions are issued by the processor (figure 2, 204-0] **using an address** [address bus, figure 4] **in the memory to identify a first block operand, the cache accumulator memory is configured to output the first block operand from the plurality of block storage locations to the functional unit** [figure 4, data path from the OUT port of the redundancy accumulator memory to a latch (306), then to the accumulator circuit (305)]; **and**

wherein the plurality of block storage locations are further configured to accumulate an intermediate result of a block accumulation operation performed on the first block operand [figure 4 shows the redundancy accumulation operations; the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57); figure 4, data path from the output of the accumulator circuit (305) to a data selector (304), to a multiplexer (310), and to the IN port of the redundancy accumulator memory; column 7, lines 52-60], **wherein the intermediate result is both a result of and an operand of the block accumulation**

operation [figure 4 shows the redundancy accumulation operations; the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57); Crater et al. teach that the operand for the redundancy calculation may come from the intermediate results stored in the redundancy accumulator memory: the multiplexers shown in figure 4, 304 & 310 select an operand from either an operand supplied from the external memory via the data input bus (the corresponding data path is DATA INPUT BUS → LATCH (303) →DATA SELECTOR (304) → multiplexer (306) → IN port of the redundancy accumulator (301)), or an operand provided by the redundancy accumulator (the corresponding data path is OUT port of the redundancy accumulator (301) → LATCH (306) → multiplexer (306) → IN port of the redundancy accumulator (301)); in the case where data is read from redundancy accumulator memory ... (column 8, lines 65-67; column 9, lines 21-36; column 9, lines 49-68; column 10, lines 1-9)] **such that during the first accumulation operation, the plurality of block storage locations are concurrently configured both to cache certain ones of the block operands and to accumulate the immediate result of the first accumulation operation** [When valid prior data or prior redundancy products are stored in the addressed memory location of redundancy accumulator memory 301, these are used by redundancy generator 305 to combine with the received data byte from latch 303 in the present physical track to generate a new redundancy product which is written into redundancy accumulator memory 301 as part of the well known redundancy calculation. As each data byte is read from the presently received physical track and transmitted through latch 303 to redundancy generator 305 and multiplexer 304, this data is either directly written into the addressed memory location in redundancy accumulator memory 301 or used to perform a redundancy calculation with the prior product stored therein and this new product is then written into the addressed memory location (column 9, lines 55-63); This requires the use of redundancy accumulator memory 301 which is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 53-57)].

As to claim 33, Crater et al. disclose **a method of performing a block accumulation operation** [Redundancy Accumulator for Disk Drive Array Memory (title)], **the method comprising**:

storing data to a storage array including a plurality of mass storage devices [data storage subsystem (figure 1, 100) comprising a plurality of disk drives (figure1, 103-I)];

receiving a first command to perform a block accumulation operation on a first block operand identified by a first address in a memory [instructions are issued by the processor (figure 2, 204-0) to perform redundancy accumulation operation (abstract); the redundancy accumulator memory accumulate data over a number of data elements, such as records, sectors or tracks (column 7, lines 42-45), hence performing block operations], **wherein the first block operand corresponds to data**

stored to the storage array [the redundancy accumulator memory accumulate data over a number of data elements, such as records, sectors or tracks (column 7, lines 42-45), hence performing block operations; the data transmitted by the associated computer system is used to generate redundancy information (abstract)], **and wherein the memory** [the corresponding memory is the memory elements (figure 3, 340~355; column 7, lines 5-20) of the cache memory unit shown in figure 1, 113, figure 2, 113 and figure 3] **configured to provided an addressable block operand storage space and to store block operands within the block operand storage space block operands received from one or more of said plurality of mass storage devices** [the memory elements (figure 3, 340~355; column 7, lines 5-20) serves as a cache (figure 1, 113) between a mass memory entity comprising a plurality of storage devices (disk drive units, figure 1, 122-l~125-r), and a plurality of host processors (figure 1, 11~12), and all data transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory (column 6, 7-10) to provide the operands of the redundancy accumulation calculations];

in response to receiving the first command:

loading the first block operand from the memory into one of a plurality of block storage locations included in a cache accumulator memory [the corresponding cache accumulator memory is the redundancy accumulator memory (figure 4, 301), which resides inside the cache controller (figure 3, 331 and 332) and serves as a cache between a plurality of cache memory elements (figure 3, 340~355) and a plurality of interface units (figure 3, 361~364). This is the case because one of the main object of Crater et al.'s invention is to provide reliability by performing redundancy calculation of the data to form a redundancy group in the disk drive subsets. To achieve this goal, not only that all data-transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory 113 (column 6, 7-10), the data must also go through the redundancy accumulator memory 301 (figures 3 and 4) so that the parity can be calculated [the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57)]. In other words, the redundancy accumulator memory (figure 4, 301) serves as a cache of cache memory elements (figure 3, 340~355); the redundancy accumulator memory is typically an n by k memory (column 8, lines 23-24), hence providing a plurality of block storage locations] **if the first block operand is not stored in the cache accumulator memory** [figure 4, the first block operand is loaded into the cache accumulator (301) from the DATA INPUT BUS via a latch (303), a data selector (304), a multiplexer (310) and the IN port of the cache accumulator; the ADDRESS BUS provides the address information], **wherein the plurality of block storage locations are configured to cache a portion of the block operand storage space of the memory** [the redundancy accumulator memory (figure 4, 301), which resides inside the cache controller (figure 3, 331 and 332) and serves as a cache between a plurality of cache memory elements (figure 3, 340~355) and a plurality of interface units (figure 3, 361~364). This is the case because one of the main object of Crater et al.'s invention is to provide reliability by performing redundancy calculation of the data to form a redundancy group in the disk drive subsets. To achieve this goal, not only that all data-transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory 113 (column 6, 7-10), the data must also go through the redundancy accumulator memory 301 (figures 3 and 4) so that the parity can be calculated [the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57)]. In other words, the redundancy accumulator memory (figure 4, 301) serves as a cache of cache memory elements (figure 3, 340~355)];

providing the first block operand from the plurality of block storage locations of the cache accumulator memory to a functional unit [the redundancy accumulator memory is typically an n by k memory (column 8, lines 23-24), hence providing a plurality of block storage locations; figure 4, the first block operand is provided by the cache accumulator using the data path from the OUT port of the redundancy accumulator memory to a latch (306), then to the accumulator circuit (305)]; **and**

accumulating a block result of the block accumulation operation generated by the functional unit into the plurality of block storage locations [figure 4, the result of the block accumulator is stored in the cache accumulator using the data path from the output of the accumulator circuit (305) to a data selector (304), to a multiplexer (310), and to the IN port of the redundancy accumulator memory; column 7, lines 52-60], **wherein the block result is both a result of and an operand of the block accumulation operation** [figure 4; column 9, lines 1-68; column 10, lines 1-9; the cache accumulator memory (figure 3, 113), including the memory elements (figure 3, 340~355) and the cache memory controllers (figure 3, 331~332) where the redundancy accumulator memory resides (figure 4, 301); figure 4 shows the redundancy accumulation operations; the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57); Crater et al. teach that the operand for the redundancy calculation may come from the intermediate results stored in the redundancy accumulator memory: the multiplexers shown in figure 4, 304 & 310 select an operand from either an operand supplied from the external memory via the data input bus (the corresponding data path is DATA INPUT BUS → LATCH (303) →DATA SELECTOR (304) → multiplexer (306) → IN port of the redundancy accumulator (301)), or an operand provided by the redundancy accumulator (the corresponding data path is OUT port of the redundancy accumulator (301) → LATCH (306) → multiplexer (306) → IN port of the redundancy accumulator (301)); in the case where data is read from redundancy accumulator memory ... (column 8, lines 65-67; column 9, lines 21-36; column 9, lines 49-68; column 10, lines 1-9)], **such that during the first accumulation operation, the plurality of block storage locations are concurrently configured both to cache certain ones of the block operands and to accumulate the immediate result of the first accumulation operation** [When valid prior data or prior redundancy products are stored in the addressed memory location of redundancy accumulator memory 301, these are used by redundancy generator 305 to combine with the received data byte from latch 303 in the present physical track to generate a new redundancy product which is written into redundancy accumulator memory 301 as part of the well known redundancy calculation. As each data byte is read from the presently received physical track and transmitted through latch 303 to redundancy generator 305 and multiplexer 304, this data is either directly written into the addressed memory location in redundancy accumulator memory 301 or used to perform a redundancy calculation with the prior product stored therein and this new product is then written into the addressed memory location (column 9, lines 55-63); This requires the use of redundancy accumulator memory 301 which is used to <u>store the intermediate result</u> of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 53-57)].

As to claim 34, Crater et al. disclose **an apparatus** [Redundancy Accumulator for Disk Drive Array Memory (title)], **comprising**:

storage array means configured for storing data [data storage subsystem (figure 1, 100) comprising a plurality of disk drives (figure1, 103-I); **and**

means for performing block operations on data stored to the storage array means [figure 4 shows the circuits for performing block redundancy accumulation operation] **wherein the means for performing block operations is configured to generate block results** [the redundancy accumulator memory accumulate data over a number of data elements, such as records, sectors or tracks (column 7, lines 42-45), hence performing block operations]; **and**

means for accumulating block results generated by the means for performing block operations [figure 4 shows the circuits for performing block redundancy accumulation operation], **wherein the means for accumulating block results comprises a plurality of block storage locations configured to cache a portion of the block operand storage space of the memory** [the redundancy accumulator memory (figure 4, 301), which resides inside the cache controller (figure 3, 331 and 332) and serves as a cache between a plurality of cache memory elements (figure 3, 340~355) and a plurality of interface units (figure 3, 361~364). This is the case because one of the main object of Crater et al.'s invention is to provide reliability by performing redundancy calculation of the data to form a redundancy group in the disk drive subsets. To achieve this goal, <u>not only that all data-transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory</u>

113 (column 6, 7-10), the data must also go through the redundancy accumulator memory 301 (figures 3 and 4) so that the parity can be calculated [the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57)]. In other words, the redundancy accumulator memory (figure 4, 301) serves as a cache of cache memory elements (figure 3, 340~355)] **of a memory** [the corresponding memory is the memory elements (figure 3, 340~355; column 7, lines 5-20) of the cache memory unit shown in figure 1, 113, figure 2, 113 and figure 3] **configured to store within the block operand storage space block operands received from the storage array means** [the memory elements (figure 3, 340~355; column 7, lines 5-20) serves as a cache (figure 1, 113) between a mass memory entity comprising a plurality of storage devices (disk drive units, figure 1, 122-l~125-r), and a plurality of host processors (figure 1, 11~12), and all data transfers between a host processor and a redundancy group in the disk drive subsets are routed through cache memory (column 6, 7-10) to provide the operands of the redundancy accumulation calculations] **and further configured to provide a block operand to the means for performing a first block operation in response to an instruction that uses an address in the memory to identify a first block operand** [figure 4, the first block operand is provided by the cache accumulator using the data path from the OUT port of

the redundancy accumulator memory to a latch (306), then to the accumulator circuit (305); the ADDRESS BUS provides the address information], **wherein the means for storing the block result are coupled to the means for storing the block result and the means for performing a block operation** [figure 4];

wherein the plurality of block storage locations are configured to accumulate an intermediate result of a block accumulation operation performed on the first block operand [figure 4 shows the redundancy accumulation operations; the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to <u>store the intermediate result of the redundancy calculations</u> until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57); figure 4, data path from the output of the accumulator circuit (305) to a data selector (304), to a multiplexer (310), and to the IN port of the redundancy accumulator memory; column 7, lines 52-60], **wherein the intermediate result is both a result of and an operand of the block accumulation operation** [figure 4 shows the redundancy accumulation operations; the data elements, such as physical tracks are moved one at a time to the redundancy generator (figure 4) as they are sent via the optical file backend channels to the data storage elements in a redundancy group. This requires the use of redundancy accumulator memory (figure 4, 301) which, is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 49-57); Crater et al. teach that the operand for the redundancy calculation may come from the intermediate results stored in the redundancy accumulator memory: the multiplexers shown in figure 4, 304 & 310 select an operand from either an operand supplied from the external memory via the data input bus (the corresponding data path is DATA INPUT BUS → LATCH (303) →DATA SELECTOR (304) → multiplexer (306) → IN port of the redundancy accumulator (301)), or an operand provided by the redundancy accumulator (the corresponding data path is OUT port of the redundancy accumulator (301) → LATCH (306) → multiplexer (306) → IN port of the redundancy accumulator (301)); in the case where data is read from redundancy accumulator memory ... (column 8, lines 65-67; column 9, lines 21-36; column 9, lines 49-68; column 10, lines 1-9)], **such that during the first accumulation operation, the plurality of block storage locations are concurrently configured both to cache certain ones of the block operands and to accumulate the immediate result of the first accumulation operation** [When valid prior data or prior redundancy products are stored in the addressed memory location of redundancy accumulator memory 301, these are used by redundancy generator 305 to combine with the received data byte from latch 303 in the present physical track to generate a new redundancy product which is written into redundancy accumulator memory 301 as part of the well known redundancy calculation. As each data byte is read from the presently received physical track and transmitted through latch 303 to redundancy generator 305 and multiplexer 304, this data is either directly written into the addressed memory location in redundancy accumulator memory 301 or used to perform a redundancy calculation with the prior product stored therein and this new product is then written into the addressed memory location (column 9, lines 55-63); This requires the use of redundancy accumulator memory 301 which is used to store the intermediate result of the redundancy calculations until all of the physical tracks have been included in the redundancy calculation (column 7, lines 53-57)].

As to claim 2, Crater et al. teach that **the cache accumulator memory** [the corresponding cache accumulator memory is the redundancy accumulator memory (figure 4, 301), which resides inside the cache controller (figure 3, 331 and 332) and serves as a cache between a plurality of cache memory elements (figure 3, 340~355) and a plurality of interface units (figure 3, 361~364)] **comprises a dual-ported memory** [figure 4 shows that the redundancy accumulator memory (301) has two ports: IN and OUT].

As to claim 4, Crater et al. do not explicitly mention that **the cache accumulator is configured to indicate whether a particular block operand stored in the cache accumulator is modified with respect to a copy of that particular block operand in the memory**. However, it is inherent for all cache memory systems that a mechanism is required to maintain data coherency between the main memory and the cache memory, and as such an indicator, commonly known as the "dirty bit," is required to indicate whether the data in the cache has been modified and hence is different from the corresponding copy in the main memory. Therefore, this claim is anticipated by the invention of Crater et al.

As to claim 5, Crater et al. disclose that **the cache accumulator memory is configured to load a copy of the block operand into the cache accumulator memory from the memory in response to the block operand not being present in the cache accumulator memory when the instruction is received** [figure 4, the first block operand is loaded into the cache accumulator (301) from the DATA INPUT BUS via a latch (303), a data selector (304), a multiplexer (310) and the IN port of the cache accumulator; the ADDRESS BUS provides the address information].

As to claim 6, a **replacement policy**, which overwrites a location in the cache with new data to be copied into the cache when all locations in cache are currently storing valid data, is an inherent property of any cache system.

As to claim 9, Crater et al. teach that **the functional unit** [the redundancy accumulation calculator, figure 4, 305] **is configured to perform a parity calculation** [the redundancy data generations such as simple parity, orthogonal parity, or Reed-Solomon code (column 7, lines 38-41)] **on the block operand** [the redundancy accumulator memory accumulate data over a number of data elements, such as records, sectors or tracks (column 7, lines 42-45), hence performing block operations].

As to claim 11, Crater et al. teach that **the functional unit** [the redundancy accumulation calculator, figure 4, 305] **is configured to calculate a parity block** [the redundancy data generations such as simple parity, orthogonal parity, or Reed-Solomon code (column 7, lines 38-41)] **from a plurality of data blocks in a stripe of data, wherein the first block operand is a first one of the data blocks in the stripe of data** [the data transmitted by the associated computer system is used to generate redundancy information which is written with the data across N+M disk drives in a redundancy group in the data storage subsystem (abstract)].

As to claim 12, Crater et al. disclose that **the functional unit is configured to perform the operation on two block operands** [figure 4 shows that the redundancy accumulation calculator (305) has two input blocks, one from latch (303) and the second from another latch (306)].

As to claim 13, Crater et al. disclose that **a first of the two block-operands is the first block operand stored in the cache accumulator memory** [figure 4, the first block operand is provided by the cache accumulator using the data path from the OUT port of the redundancy accumulator memory to a latch (306), then to the accumulator circuit (305)] **and a second of the two block-operands is provided on a data bus** [figure 4, the second block operand is loaded into the cache accumulator (301) from the DATA INPUT BUS via a latch (303), a data selector (304), a multiplexer (310) and the IN port of the cache accumulator; the ADDRESS BUS provides the address information], **coupled to provide operands to the functional unit** [figure 4, 305].

As to claim 14, refer to "As to claim 13." It should be noted that the second operand in this case is provided from the memory to the functional unit via the DATA INPUT BUS (figure 4).

As to claim 15, Crater et al. teach that **the cache accumulator memory is configured to provide a word of the block operand to the functional unit during an access cycle in which cache accumulator also stores a word of the block result generated by the functional unit** [figure 4, the first block operand is provided by the cache accumulator using the data path from the OUT port of the redundancy accumulator memory to a latch (306), then to the accumulator circuit (305); figure 4, the result of the block accumulator is stored in the cache accumulator using the data path from the output of the accumulator circuit (305) to a data selector (304), to a multiplexer (310), and to the IN port of the redundancy accumulator memory; column 7, lines 52-60].

As to claim 17, refer to "As to claim 32," "As to claim 33" and "As to claim 34."

As to claim 18, refer to "As to claim 2."

As to claim 20, refer to "As to claim 6."

As to claim 23, refer to "As to claim 9."

As to claim 24, refer to "As to claim 1" and "As to claim 9."

As to claim 25, refer to "As to claim 12."

As to claim 26, refer to "As to claim 13."

As to claim 28, refer to "As to claim 1."

As to claim 29, refer to "As to claim 13" and "As to claim 9."

As to claim 30, refer to "As to claim 11."

As to claim 31, refer to "As to claim 34."

8. *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 3 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crater et al., (US 5,146,588), and in view of Faraboschi et al. (U.S. 6,122,708).

As to claims 3 and 19, Crater et al. do not mention that **the cache accumulator memory comprises at least two independently interfaced memory banks**.

However, Faraboschi et al. discloses a data cache system for use with streaming data in which the data cache consists of two independently interfaced memory banks (figure 3, items 130 and 132), that The data cache memory may include a single bank, or two or more banks in a set associative configuration, with each bank includes a data cache, a tag array, and addressing circuitry (column 3, lines 47-50).

Two-bank organization of the cache system allows data to be transferred to and from the cache system simultaneously using the two banks, such as **providing the block operand from a first storage location in a first one of the independently interfaced memory banks and to store the block result in a second block storage location in a second one of the independently interfaced memory banks** (this is the case where a vector/block extends across one or more block boundaries explained earlier), hence avoiding the situation where a single-bank cache becomes the bottleneck of memory access and will reduce the overall memory access latency.

Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the benefits offered by a two-bank cache memory architecture and to adopt it for the cache unit in the apparatus disclosed by Crater et al. to further improve its performance.

10. Claims 7-8 and 21-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Crater et al., (US 5,146,588), and in view of Handy, "The cache memory book: the authoritative reference on cache design," Academic Press, 1993, page 57.

As to claim 7, Crater et al. do not explicitly mention that **the cache accumulator is configured to use a least recently used algorithm to select the first set of block storage locations to overwrite**, since the disclosure focuses on the aspect of vector processing using a data cache.

However, Handy teaches that a replacement algorithm is required in a cache system to select which entry in the cache is to be replaced when a new line is to be brought into the cache, and that the least recent used algorithm is one of the most commonly adopted scheme.

Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the need to have a replacement algorithm and the benefit offered by a least recently used algorithm and to adopt it for the cache unit in the apparatus disclosed by Crater et al.

As to claim 8, Crater et al. do not explicitly mention that if data in the selected one of the block storage locations is modified with respect to a copy of that data in the memory, the cache accumulator is configured to write the data back to the memory before loading the copy of the block operand into the selected one of the block storage locations, since the disclosure focuses on the aspect of vector processing using a data cache.

However, Handy teaches that a write strategy is required in a cache system to deal with the situations where data is modified in either he cache or the main memory, which leads to data inconsistency between the main memory and cache. Particularly, Handy teaches that a technique, known as "write –through," in which the main memory is always updated first during all write cycles, is commonly adopted in cache system design (pages 64-65). With such a write-through policy, data consistency between the main memory and the cache will be enforced.

Therefore it would have been obvious for persons of ordinary skills in the art at the time of applicant's invention to recognize the need to have a write policy and the benefit offered by the write-through algorithm, and to adopt it for the cache unit in the apparatus disclosed by Crater et al.

As to claim 21, refer to "As to claim 7."

As to claim 22, refer to "As to claim 8."

Conclusion

11. Claims 2-9, 11-15, 17-26 and 28-34 are rejected as explained above.

12. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sheng-Jen Tsai
Examiner
Art Unit 2186

June 9, 2006

PIERRE BATAILLE
PRIMARY EXAMINER

6\15\06